Decentralized Fault Prognosis of Discrete-Event Systems Using State-Estimate-Based Protocols

Xiang Yin, *Member, IEEE*, and Zhaojian Li, *Member, IEEE*

***Abstract*—We investigate the problem of decentralized fault prognosis in the context of discrete-event systems. In this problem, the system is monitored by a set of local agents; each of them sends its local information to a coordinator in order to issue a fault alarm before the occurrence of fault. Two new decentralized protocols are proposed by exploiting the state-estimate of each local agent. For each protocol, a necessary and sufficient condition for its correctness is proposed; they are termed as positive state-estimate-prognosability and negative state-estimate prognosability. Verification algorithms for the necessary and sufficient conditions are also provided. We show that the proposed new protocols are incomparable with any of the existing protocols in the literature. Therefore, they provide new opportunities for correctly predicting the fault when all existing protocols fail.**

***Index Terms*—Complexity, decentralized fault prognosis, discrete-event systems (DESs), state-estimate.**

I. Introduction

FAULT detection and prediction are crucial tasks in complex automated systems. In many *safety-critical systems*, when the system is subject to fault, simply detecting and isolating the fault may not be enough to guarantee the safety of the system, since some critical functionality of the system can be destroyed before the detection of fault. Therefore, in some applications, it is of interest to *predict* the occurrence of fault in advance such that some protective actions can be taken before fault occurs. This problem is referred to as the fault prognosis (or prediction) problem.

In this paper, we consider the fault prognosis problem in the context of discrete-event systems (DES). DESs is a class of dynamic systems with discrete state spaces and event-triggered dynamics [3]. In the past two decades, the theory of model-based fault diagnosis using DES model has been extensively developed and has been successfully applied to many applications; see [6], [19], [20], [22], [26], [27] and a recent survey [40]. More recently, the problem of fault prognosis has draw considerable attention in DES literature, see [1], [2], [4], [5], [7], [10], [11], [15], [16], [18], [21], [23], [24], [28]–[32], [34], [36], [37], [39]. The problem of fault prognosis of DES was initially studied by [10] and [11], where the notion of predictability (or uniformly bounded prognosability) was introduced. In [18], using the concept of indicator strings, the notion of prognosability was introduced as the necessary and sufficient condition for the existence of a prognoser that can correctly predict the fault occurrences. Later on, in the context of DES, the fault prognosis problem has been further studied for stochastic systems [2], [5], [7], [23], [24], distributed systems [30], [31], timed systems [4], and Petri nets [1], [21]. The problem of enforcing prognosability by activating/deactivating sensors is studied in [34].

In many large-scale networked systems, the information structure of the system is naturally decentralized due to the distributed physical components of the system. It is impossible or very costly to collect all available data at a central station. Therefore, using decentralized architecture to process data is more efficient for large-scale systems. More specifically, we assume that the system is monitored by several *local agents*; each of them has its own observation and data processing capability. Each local agent first processes the data it observes locally and then sends the processed and compressed data to a central fusion site (or coordinator). Then the coordinator uses the local information received to issue a global prognostic decision. Following this decentralized scheme, the problem of decentralized fault prognosis has been studied by many works in the DES literature; see [16], [18], [29], [36], [37].

In the decentralized fault prognosis problem, one of the key ingredients is what protocol (or architecture) we adopt to handle the decentralized information. Roughly speaking, a decentralized protocol consists of: 1) what information each local agent sends to the coordinator and 2) how the coordinator issues a global decision based on the local information received. For example, Kumar and Takai [18] assumed that each local agents can only send a binary information and use disjunctive rule for the coordinator. Similarly, Khoumsi and Chakib [16] still considered using binary information but with conjunctive fusion rule. Using inference-based fusion rule for decentralized fault prognosis is considered in [29].

In this paper, we propose two new decentralized protocols for the purpose of fault prognosis; namely, the positive state-estimate (PSE) based protocol and the negative state-estimate (NSE) based protocol. In both of these two protocols, each

Manuscript received February 4, 2017; revised August 17, 2017 and November 18, 2017; accepted January 22, 2018. Date of publication February 12, 2018; date of current version February 22, 2019. This paper was recommended by Associate Editor P. Chen. *(Corresponding author: Xiang Yin.)*

X. Yin is with the Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xiangyin@umich.edu).

Z. Li is with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: lizhaoj1@msu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2799961

local agent uses a subset of its *state-estimate* as the information it sends to the coordinator. More specifically, in the PSE-based protocol, each local agent sends the set of states, which is believed as the reason why a fault alarm should be issued; while in the NSE-based protocol, each local agent sends the set of states, which is believed as the reason why a fault alarm *should not* be issued. Then the coordinator takes the intersection of the local state-estimates to calculate a global prognostic decision.

The main contributions of this paper are as follows. First, we formally define the PSE-based protocol and the NSE-based protocol; the implementation issues of these two protocols are discussed. Second, for each protocol, we provide a necessary and sufficient condition under which the protocol satisfies the following two requirements: 1) a fault alarm can be issued K steps before any fault occurrence and 2) once a fault alarm is issued, a fault is guaranteed to occur within M steps. The necessary and sufficient condition is termed as PSE-prognosability (respectively, NSE-prognosability) for the positive (respectively, negative) state-estimate-based protocol. Third, effective algorithms for the verification of PSE-prognosability and NES-prognosability are provided, respectively; the former has a polynomial complexity with respect to the size of the system, while the latter has an exponential complexity with respect to the size of the system. Moreover, we show that the verification of both of PSE-prognosability and NSE-prognosability are PSPACE-hard with respect to the number of local agents. Finally, we show that the proposed protocols are *incomparable* with any existing decentralized prognosis protocol in the literature. This also justifies the usefulness of the proposed protocols, since they provide new opportunities for predicting fault when all existing protocols fail to do so.

It is worth remarking that state-estimate-based protocols have been used in the literature for the purposes of decentralized fault diagnosis [9], [12], [25] and decentralized control [35]. More generally, the idea of taking the intersection of local (language) information has also been explored in [13] and [14] for the purpose of supervisory control. The differences between the results in this paper and the above works are as follows. First, in this paper, we consider the decentralized fault prognosis problem, which is different from the decentralized fault diagnosis problem and the decentralized control problem. Although there is a connection between decentralized fault diagnosis and decentralized control [33], to the best of our knowledge, there is no formal relationship between the decentralized fault prognosis problem and these two problems established in the literature. Therefore, it is needed and nontrivial to develop state-estimate-based protocol for the purpose of fault prognosis. Second, the protocols in [9], [12], [25], and [35] are more close to the positive protocol in our setting. However, we consider both positive and negative protocols in this paper. In particular, the necessary and sufficient condition of the negative protocol is more difficult to verify in the prognosis problem, since we not only need to consider the current information, but also need to consider the previous information along the trajectory. This issue is similar to that the disjunctive architecture versus the conjunctive architecture under the binary information setting [38].

The remaining part of this paper is organized as follows. Section II provides some necessary preliminaries for fault prognosis of DES. In Section III, we propose two novel state-estimate-based protocols for the purpose of decentralized fault prognosis. The notion of PSE-prognosability and NSE-prognosability are introduced. Verification algorithms for PSE-prognosability and NSE-prognosability are provided in Section IV. Section V discusses some complexity issues in the verification problems. In Section VI, we compare the proposed protocols with the existing ones in the literature. Finally, we conclude this paper in Section VII.

II. Preliminary

A. Discrete Event Systems

Let Σ be a finite set of events. A string $s = \sigma_1 \cdots \sigma_n, \sigma_i \in \Sigma$ is a finite sequence of events. We denote by Σ^* the set of strings over Σ including the empty string ϵ. For any string s, we denote by $|s|$ its length with $|\epsilon| = 0$. A language $L \subseteq \Sigma^*$ is a set of strings. We denote by $\overline{L}$ the prefix-closure of L, i.e., $\overline{L} = \{s \in \Sigma^* : \exists t \in \Sigma^* \text{ s.t. } st \in L\}$. For any $s \in \Sigma^*$, we denote by L/s the set of continuations of s in L, i.e., $L/s = \{t \in \Sigma^* : st \in L\}$.

A DES is modeled by a finite-state automaton

$$G = (Q, \Sigma, \delta, q_0, Q_m) \tag{1}$$

where Q is the finite set of states, $Q_m \subseteq Q$ is the set of marked states, Σ is the finite set of events, $q_0 \in Q$ is the initial state, and $\delta : Q \times \Sigma \to Q$ is the partial transition function, where for any $q, q' \in Q$ and $\sigma \in \Sigma$, $\delta(q, \sigma) = q'$ means that there exists a transition from q to q' labeled with σ. Function δ is also extended to $\delta : Q \times \Sigma^* \to Q$ recursively by: for any $q \in Q, s \in \Sigma^*, \sigma \in \Sigma$, $\delta(q, s\sigma) = \delta(\delta(q, s), \sigma)$. For the sake of simplicity, we also write $\delta(q, s)$ as $\delta(s)$ if $q = q_0$. The language generated by G from state $q \in Q$ is $\mathcal{L}(G, q) = \{s \in \Sigma^* : \delta(q, s)!\}$, where "!" means "is defined." Then the language generated by G is $\mathcal{L}(G) := \mathcal{L}(G, q_0)$. The language marked by G is $\mathcal{L}_m(G) = \{s \in \Sigma^* : \delta(s) \in Q_m\}$. We will omit Q_m and write an automaton by $G = (Q, \Sigma, \delta, q_0)$ if marking is not considered. We say that G is *live* if $\forall q \in Q, \exists \sigma \in \Sigma : \delta(q, \sigma)!$. Hereafter, we assume without loss of generality (w.l.o.g.) that G is live.

For any two automata $A = (Q_A, \Sigma, \delta_A, q_{0,A})$ and $B = (Q_B, \Sigma, \delta_B, q_{0,B})$ such that $\mathcal{L}(A) \subseteq \mathcal{L}(B)$, we say that A is a subautomaton of B, denoted by $A \sqsubseteq B$, if: 1) $Q_A \subseteq Q_B$; 2) $q_{0,A} = q_{0,B}$; and 3) $\forall s \in \mathcal{L}(B) : \delta_A(s) = \delta_B(s)$ if $\delta_A(s)!$. We say that A is a strict subautomaton of B, denoted by $A \sqsubset B$, if: 1) $A \sqsubseteq B$ and 2) $\forall s \in \mathcal{L}(B) \setminus \mathcal{L}(A) : \delta_B(s) \in Q_B \setminus Q_A$. Note that, for any A and B such that $\mathcal{L}(A) \subseteq \mathcal{L}(B)$, we can always refine their state-spaces in polynomial-time such that $A \sqsubset B$; see [8].

B. Decentralized Fault Prognosis

In the fault prognosis problem, we want to predict any fault *before* its occurrence. To this end, we denote by $H = (Q_H, \Sigma, \delta_H, q_{0,H})$ the specification automaton that models the *normal behaviors* of the system, i.e., any string in $\mathcal{L}(H) \subseteq \mathcal{L}(G)$ is considered as a nonfault string and any string in

$\mathcal{L}(G) \setminus \mathcal{L}(H)$ is considered as a fault string. Note that we do not consider marking in H and G since the fault prognosis problem is studied on the prefix-closed language generated by the system. w.l.o.g, we assume that $H \sqsubset G$. With this assumption, for any string $s \in \mathcal{L}(G)$, s is a nonfault string iff $\delta(s) \in Q_H$, i.e., Q_H is the set of nonfault states and $Q \setminus Q_H$ is the set of fault states.

In the decentralized fault prognosis problem, the plant G is monitored by a set of local agents (or local prognosers). We assume that there are n local agents and denote by $\mathcal{I} = \{1, \ldots, n\}$ the index set. For each agent $i \in \mathcal{I}$, we denote by $\Sigma_{o,i}$ the set of events that can be observed locally by i. Then $P_i : \Sigma^* \to \Sigma_{o,i}^*$ is the natural projection defined by

$$P_i(\epsilon) = \epsilon \quad \text{and} \quad P_i(s\sigma) = \begin{cases} P_i(s)\sigma & \text{if } \sigma \in \Sigma_{o,i} \\ P_i(s) & \text{if } \sigma \notin \Sigma_{o,i}. \end{cases}$$

Function P_i is also extended 2^{Σ^*} by $\forall L \in 2^{\Sigma^*} : P_i(L) = \{s \in \Sigma_{o,i}^* : \exists t \in L \text{ s.t. } P_i(t) = s\}$. We denote by P_i^{-1} the inverse projection of P_i.

The basic scheme of the decentralized prognosis is as follows. At each instant, each local agent, based its own observation, sends a *local information* to the *coordinator*. Then the coordinator computes a *global decision* based on the local information received. More specifically, each local prognoser $i \in \mathcal{I}$ is a function

$$\mathcal{D}_i : P_i(\mathcal{L}(G)) \to \mathcal{A} \tag{2}$$

where $\mathcal{A}$ is the set of symbols it can send to the coordinator, i.e., the space of communicating information. Then the coordinator is a (memoryless) function

$$\mathcal{C} : \mathcal{A} \times \cdots \times \mathcal{A} \to \{0, 1\} \tag{3}$$

where global decision "1" implies that a fault alarm is issued and global decision "0" means that no fault alarm is issued. The memoryless constraint essentially requires that the fusion site can be easily implemented via simple memoryless devices. This also corresponds to the essence of the decentralized decision making problem, i.e., most of the useful information need to be processed locally before sending to the coordinator. With local decision functions $\mathcal{D}_i$ and the coordinator function $\mathcal{C}$, we can also write the overall decentralized prognosis system as a function $\{\mathcal{D}_i\}_{i\in\mathcal{I}} : \mathcal{L}(G) \to \{0, 1\}$ such that

$$\forall s \in \mathcal{L}(G) : \{\mathcal{D}_i\}_{i\in\mathcal{I}}(s) = \mathcal{C}(\mathcal{D}_1(P_1(s)), \ldots, \mathcal{D}_n(P_n(s))) \tag{4}$$

we also refer to $\{\mathcal{D}_i\}_{i\in\mathcal{I}}$ as the decentralized prognoser.

In order to make sure that the decentralized prognoser works "correctly," we need to put some requirements for $\{\mathcal{D}_i\}_{i\in\mathcal{I}}$. In this paper, we consider the following two criteria proposed in [36], in order to evaluate the performance of a decentralized prognoser.

1) Any fault can be predicted K steps before it occurs, i.e., for any fault string $s \in \mathcal{L}(G) \setminus \mathcal{L}(H)$, we have

$$\big(\exists vu \in \overline{\{s\}} : vu \in \mathcal{L}(H) \wedge |u| \geq K\big)\big[\{\mathcal{D}_i\}_{i\in\mathcal{I}}(v) = 1\big]. \tag{5}$$

2) Once a fault alarm is issued, a fault is guaranteed to occur within M steps, i.e., for any string $s \in \mathcal{L}(H)$, we have

$$\{\mathcal{D}_i\}_{i\in\mathcal{I}}(s) = 1 \Rightarrow (\forall t \in \mathcal{L}(G)/s : |t| \geq M)\big[st \notin \mathcal{L}(H)\big]. \tag{6}$$

Hereafter, we will also refer to (M, K) are the *performance bound* of the prognosis system. Note that these criteria generalize the criteria in [18], which are special cases of the above ones by taking $K = 0$ and $M = |Q|$.

Remark 1: The above definition of $\{\mathcal{D}_i\}_{i\in\mathcal{I}}$ is generic; it remains to specify functions $\mathcal{D}_i, \mathcal{C}$ and set $\mathcal{A}$ in order to specifically define a decentralized prognoser. In fact, the choice of $\mathcal{D}_i, \mathcal{A}$ and $\mathcal{C}$ is referred to as the *architecture/protocol* of the decentralized system. By choosing different $\mathcal{D}_i, \mathcal{C}$ and $\mathcal{A}$, different architectures/protocols have been proposed in the literature. For example, in the disjunctive architecture [18] $\mathcal{A}$ is chosen to be $\{0, 1\}$ and function $\mathcal{C}$ is the disjunction of the local binary values. In the inference-based architecture [29], $\mathcal{A}$ is chosen to be $\{0, 1, \phi\} \times \{0, 1, \ldots, N\}$, where the first component represents local decisions and the second component represents ambiguity levels, and $\mathcal{C}$ simply selects the local decision with the smallest ambiguity level. In this paper, we will present a new decentralized protocol, where $\mathcal{A}$ is a set of states and $\mathcal{C}$ is a function that involves set intersection. We will elaborate on this next.

III. State Estimate-Based Protocol

When string $s \in \mathcal{L}(G)$ is generated by the system, each agent $i \in \mathcal{I}$ can observe $P_i(s)$. Its *state-estimate* upon observing $P_i(s)$, denoted by $\mathcal{E}_i(P_i(s))$, is defined as the set of states the system could be in after observing $P_i(s)$, i.e.,

$$\mathcal{E}_i(P_i(s)) = \{q \in Q : \exists t \in \mathcal{L}(G) \text{ s.t. } P_i(t) = P_i(s) \wedge \delta(t) = q\}. \tag{7}$$

In other words, $\mathcal{E}_i(P_i(s))$ essentially summarizes agent i's knowledge about the system state. The state-estimate has the following two features. First, the domain of $\mathcal{E}_i$, i.e., 2^X, is finite. Second, $\mathcal{E}_i(P_i(s))$ can be computed recursively upon the occurrence of a new event. Therefore, we may possibly use $\mathcal{E}_i$ as the local decision function and use 2^X as the set of symbols to communicate. This leads to the basic scheme of the state-estimate-based protocol: "At each instant, each local agent computes its state-estimate and sends it to the coordinator. Then the coordinator manipulates on the local state-estimates to issue a global decision."

The above basic idea requires to send all states in $\mathcal{E}_i(P_i(s))$ at each instant. However, many states in $\mathcal{E}_i(P_i(s))$ are irrelevant for the purpose of fault prognosis. To identify which states are relevant to the prognosis problem, in particular, the performance bound (M, K), we define the followings [36]. For each state $q \in Q_H$, we denote by $d_{\min}(q)$ the minimum number of steps required such that a fault can occur from q, i.e.,

$$d_{\min}(q) = \min_{s\in\mathcal{L}(G,q)\setminus\mathcal{L}(H,q)} (|s| - 1). \tag{8}$$

Note that $d_{\min}(q)$ is undefined when fault can never happen from state q, i.e., $\mathcal{L}(G, q) \setminus \mathcal{L}(H, q) = \emptyset$. Also, for each state

$q \in Q_H$, we denote by $d_{\max}(q)$ the length of the longest nonfault string that can occur from q, i.e.,

$$d_{\max}(q) = \max_{s \in \mathcal{L}(H,q)} |s|. \tag{9}$$

Note that $d_{\max}(q) = \infty$ when an arbitrarily long nonfault string can be executed from state q. Then we define two sets of states

$$\begin{aligned} \partial_K &= \{q \in Q_H : d_{\min} = K\} \\ \Upsilon_M^{>} &= \{q \in Q_H : d_{\max} \geq M\}. \end{aligned} \tag{10}$$

As shown in [36], both sets ∂_K and $\Upsilon_M^{>}$ can be computed in polynomial-time in the size of G. Also, we assume hereafter that $d_{\min}(q_0) \geq K$; otherwise, no prognoser can achieve the requirement in (5). Finally, we note that $\Upsilon_{|Q_H|}^{>} = \Upsilon_M^{>}$ for any $M \geq |Q_H|$, since that a nonfault string with length $|Q_H|$ can occur from a state implies that a nonfault cycle can be reached from this state.

Based on the above concepts, hereafter, we propose two different state-estimate-based protocols.

A. Positive State-Estimate-Based Protocol

Suppose that the default global decision of the coordinator is 0, i.e., no fault alarm is issued. In order to issue a global fault alarm, each local agent needs to send the coordinator a set of states, which is a subset of the state-estimate, as the reason why it wants to issue an alarm. Then the coordinator simply takes the intersection of the sets of states it received. If the intersection is empty, then it means that the reasons for issuing a global alarm are not consistent; hence, the coordinator will remain the default decision 0 unchanged. If the intersection is not empty, then it means that all agents agree with some common reason for issuing a global alarm; hence, the coordinator will issue 1 globally.

Based on the above discussion, we propose the *PSE-Based Protocol* $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}$ specified as follows: each local prognoser is a function

$$\mathcal{D}_i^{\text{pos}} : P_i(\mathcal{L}(G)) \to 2^{\partial_K} \tag{11}$$

such that, for any $s \in \mathcal{L}(G)$, we have

$$\mathcal{D}_i^{\text{pos}}(P_i(s)) = \mathcal{E}_i(P_i(s)) \cap \partial_K. \tag{12}$$

Then the global decision issued by the coordinator is defined by

$$\left\{\mathcal{D}_i^{\text{pos}}\right\}_{i\in\mathcal{I}}(s) = \begin{cases} 1, & \text{if } \cap_{i\in\mathcal{I}}\, \mathcal{D}_i^{\text{pos}}(P_i(s)) \neq \emptyset \\ 0, & \text{if } \cap_{i\in\mathcal{I}}\, \mathcal{D}_i^{\text{pos}}(P_i(s)) = \emptyset. \end{cases} \tag{13}$$

Intuitively, each local prognoser computes $\mathcal{E}_i(P_i(s)) \cap \partial_K$ as its reason for issuing a fault alarm. Then the coordinator takes the intersection of $D_i(P_i(s))$ to see if they have some common reason, i.e., $\cap_{i\in\mathcal{I}} D_i(P_i(s)) \neq \emptyset$. If so, then a global fault alarm will be issued, i.e., $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(s) = 1$; otherwise, the coordinator will stay silent, i.e., $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(s) = 0$. The reason why we choose to send states in ∂_K is that, ∂_K are the set of states at which we *must* issue a global fault alarm; otherwise, the alarm will be too late, i.e., the condition in (5) may be violated. We call this protocol positive since the default decision is 0, i.e., no fault alarm. We can also define a negative protocol with default decision is 1; this will be discussed later. For the sake of simplicity, hereafter, we will also refer to the PSE-based protocol as the positive protocol.

To study whether or not $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}$ satisfies the criteria in (5) and (6), we introduce the notion of PSE-based prognosability.

Definition 1: Specification H is said to be PSE-based prognosable (PSE-prognosable) with respect to G, $\Sigma_{o,i}$, $i \in \mathcal{I}$ and (M, K) if

$$\left(\forall s \in \mathcal{L}(H) : \delta(s) \in \Upsilon_M^{>}\right)\left[\left(\bigcap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))\right) \cap \partial_K = \emptyset\right]. \tag{14}$$

Intuitively, PSE-prognosability requires that for any string that leads to a state in $\Upsilon_M^{>}$, the state-estimate of each local agent should not contain a common state in ∂_K. The following result reveals that PSE-prognosability is indeed the necessary and sufficient condition such that $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}$ satisfies (5) and (6).

Theorem 1: $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}$ satisfies (5) and (6) if and only if H is PSE-prognosable with respect to G, $\Sigma_{o,i}$, $i \in \mathcal{I}$ and (M, K).

Proof: ($\Rightarrow$) If H is not PSE-prognosable with respect to G, $\Sigma_{o,i}$, $i \in \mathcal{I}$ and (M, K), then we know that there exists a string $s \in \mathcal{L}(H)$ such that $\delta(s) \in \Upsilon_M^{>}$ and $(\bigcap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_K \neq \emptyset$. Then, by (12), we know that

$$\begin{aligned} &\cap_{i\in\mathcal{I}} \mathcal{D}_i^{\text{pos}}(P_i(s)) \\ &\quad = \cap_{i\in\mathcal{I}}(\mathcal{E}_i(P_i(s)) \cap \partial_K) = (\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_K \neq \emptyset. \end{aligned}$$

Therefore, by (13), we know that $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(s) = 1$. However, since $\delta(s) \in \Upsilon_M^{>}$, we know that there exists $t \in \mathcal{L}(G)/s$ such that $|t| \geq M$ and $st \in \mathcal{L}(H)$, i.e., a fault is not guaranteed to occur in M step. This implies that (6) does not hold.

($\Leftarrow$) Suppose that H is PSE-prognosable with respect to G, $\Sigma_{o,i}$, $i \in \mathcal{I}$ and (M, K). First, we show that $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}$ satisfies (5). For any string $s \in \mathcal{L}(G) \setminus \mathcal{L}(H)$, since we assume that $d_{\min}(q_0) \geq K$ we know that there exists a nonfault prefix $s_N \in \overline{\{s\}} \cap \mathcal{L}(H)$ such that $\delta(s_N) \in \partial_K$. Moreover, for any $i \in \mathcal{I}$, we have $\delta(s_N) \in \mathcal{E}_i(P_i(s_N))$. Therefore, we know that $\cap_{i\in\mathcal{I}} \mathcal{D}_i^{\text{pos}}(P_i(s_N)) = (\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s_N))) \cap \partial_K \neq \emptyset$, i.e., $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(s_N) = 1$. Therefore, (5) is satisfied. Next, we show that $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}$ satisfies (6) by contradiction. Assume that (6) is not satisfied, i.e., there exist $s \in \mathcal{L}(H)$ and $t \in \mathcal{L}(G)/s$ such that $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(s) = 1$, $|t| \geq M$ and $st \in \mathcal{L}(H)$. For the above string s, since $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(s) = 1$, by (12) and (13), we know that $(\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_K \neq \emptyset$. Also, by $|t| \geq M$ and $st \in \mathcal{L}(H)$, we know that $d_{\max}(\delta(s)) \geq M$, i.e., $\delta(s) \in \Upsilon_M^{>}$. However, since the system is PSE-prognosable, $\delta(s) \in \Upsilon_M^{>}$ implies that $(\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_K = \emptyset$. This is a contradiction. Therefore, we know that (6) must be satisfied. ■

We illustrate the positive protocol by the following example.

Example 1: Let us consider the system automaton G and the specification automaton H in Fig. 1. Suppose that there are two local agents, i.e., $\mathcal{I} = \{1, 2\}$, where $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. Let us consider performance bound $K = 0$ and $M = |Q_H| = 9$ and we have $\partial_K = \{8\}$ and $\Upsilon_M^{>} = \{1, 2, 3, 4, 6, 7, 9\}$. Then for all strings leading to states $1, 2, 3, 4, 6, 7,$ and 9, i.e., $\epsilon, a, b, ao, bo, aoo, boo$, we have $(\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(\epsilon))) \cap \partial_K = \{1, 2\} \cap \{1, 3\} \cap \{8\} = \emptyset$,

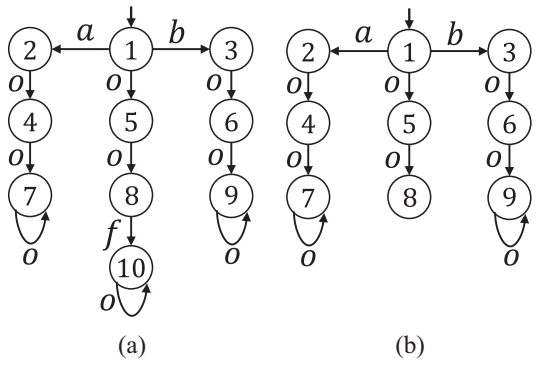


Fig. 1. H is PSE-prognosable with respect to G, $\Sigma_{o,1} = \{a, o\}$, $\Sigma_{o,2} = \{b, o\}$, $K = 0$, and $M = |Q_H|$. (a) G. (b) H.

$(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(a)))\cap\partial_K = \{2\}\cap\{1,2\}\cap\{8\} = \emptyset$, $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(b)))\cap\partial_K = \{3\}\cap\{1,3\}\cap\{8\} = \emptyset$, $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(ao)))\cap\partial_K = \{4\}\cap\{4,5\}\cap\{8\} = \emptyset$, $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(bo)))\cap\partial_K = \{6\}\cap\{5,6\}\cap\{8\} = \emptyset$, $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(aoo)))\cap\partial_K = \{7\}\cap\{7,8\}\cap\{8\} = \emptyset$, and $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(boo)))\cap\partial_K = \{9\}\cap\{8,9\}\cap\{8\} = \emptyset$. Therefore, the system is PSE-prognosable and the positive protocol $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}$ achieves the two criteria in (5) and (6). For example, let us consider fault string $oof \in \mathcal{L}(G)\setminus\mathcal{L}(H)$. For prefix $oo \in \overline{\{oof\}}$ such that $\delta(oo) = 8 \in \partial_K$, we have $\mathcal{D}_1^{\text{pos}}(oo) = \mathcal{E}_1(P_1(oo))\cap\partial_K = \{8,9\}\cap\{8\}$ and $\mathcal{D}_2^{\text{pos}}(oo) = \mathcal{E}_2(P_2(oo)) = \{7,8\}\cap\{8\}$. Since $\mathcal{D}_1^{\text{pos}}(oo)\cap\mathcal{D}_2^{\text{pos}}(oo) \neq \emptyset$, we know that $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(oo) = 1$. Therefore, a fault alarm can be issued K step before the fault occurs, i.e., (5) holds. To see that (6) holds, let us consider string $aoo \in \mathcal{L}(H)$ such that $\delta(aoo) = 7 \in \Upsilon_M^{>}$. We have $\mathcal{E}_1(P_1(aoo)) = \{7\}$ and $\mathcal{E}_2(P_2(aoo)) = \{7,8\}$. Since $\mathcal{E}_1(P_1(aoo))\cap\mathcal{E}_2(P_2(aoo))\cap\partial_K = \emptyset$, we know that $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(aoo) = 0$, i.e., no wrong fault alarm will be issued. Similarly, we can show that for any $s \in \mathcal{L}(H) : \delta(s) \in \Upsilon_M^{>}$, we have $\{\mathcal{D}_i^{\text{pos}}\}_{i\in\mathcal{I}}(s) = 0$. Therefore, the positive protocol also satisfies (6).

B. Negative State-Estimate-Based Protocol

In the above section, we have proposed the PSE-based protocol, where the default global decision is 0 and each local agent needs to tell the coordinator why it wants to issue a fault alarm. Alternatively, we can also set the default global decision as 1 and at each instant, each local agent needs to tell the coordinator why a fault alarm *should not* be issued. To this end, we propose the *NSE-Based Protocol* $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}$ defined as follows: each local prognoser is a function

$$\mathcal{D}_i^{\text{neg}} : P_i(\mathcal{L}(G)) \to 2^{\Upsilon_M^{>}} \tag{15}$$

such that, for any $s \in \mathcal{L}(G)$, we have

$$\mathcal{D}_i^{\text{neg}}(P_i(s)) = \mathcal{E}_i(P_i(s))\cap\Upsilon_M^{>}. \tag{16}$$

Then the global decision issued by the coordinator is defined by

$$\left\{\mathcal{D}_i^{\text{neg}}\right\}_{i\in\mathcal{I}}(s) = \begin{cases} 1, & \text{if } \cap_{i\in\mathcal{I}}\mathcal{D}_i^{\text{neg}}(P_i(s)) = \emptyset \\ 0, & \text{if } \cap_{i\in\mathcal{I}}\mathcal{D}_i^{\text{neg}}(P_i(s)) \neq \emptyset. \end{cases} \tag{17}$$

Intuitively, each local prognoser uses $\mathcal{E}_i(P_i(s))\cap\Upsilon_M^{>}$ as the reason why it thinks that a global fault alarm should not be issued. Then the coordinator still takes the intersection of these local reasons to determine a global decision. Note that the reason why we choose to send states in $\Upsilon_M^{>}$ is that, $\Upsilon_M^{>}$ are the set of states at which we *cannot* issue a global fault alarm; otherwise, the alarm will be too early, i.e., the condition in (6) may be violated. For the sake of simplicity, hereafter, we will also refer to the NSE-based protocol as the negative protocol.

Similar to PSE-porgnosability, to study whether or not the negative protocol satisfies the criteria in (5) and (6), we introduce the notion of NSE-based prognosability.

Definition 2: Specification H is said to be NSE-based prognosable (NSE-prognosable) with respect to G, $\Sigma_{o,i}$, $i \in \mathcal{I}$ and (M, K) if

$$(\forall s \in \mathcal{L}(H) : \delta(s) \in \partial_K)\big(\exists t \in \overline{\{s\}}\big) \times \left[\left(\bigcap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(t))\right)\cap\Upsilon_M^{>} = \emptyset\right]. \tag{18}$$

Intuitively, NSE-prognosability requires that, for any string that leads to a state in ∂_K, there must exist a prefix of this string for which the state-estimate of each local agent should not contain a common state in $\Upsilon_M^{>}$. Otherwise, a desired fault alarm cannot be issued before reaching the state in ∂_K. The following result reveals that NSE-prognosability is indeed the necessary and sufficient condition such that $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}$ satisfies (5) and (6).

Theorem 2: $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}$ satisfies (5) and (6) if and only if H is NSE-prognosable with respect to G, $\Sigma_{o,i}$, $i \in \mathcal{I}$ and (M, K).

Proof: ($\Rightarrow$) By contraposition. Suppose that H is not NSE-prognosable with respect to G, $\Sigma_{o,i}$, $i \in \mathcal{I}$ and (M, K). Then we know that there exists a string $s \in \mathcal{L}(H) : \delta(s) \in \partial_K$ such that for any $t \in \overline{\{s\}}$, we have $(\bigcap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(t)))\cap\Upsilon_M^{>} \neq \emptyset$. By (16), we know that

$$\cap_{i\in\mathcal{I}}\mathcal{D}_i^{\text{neg}}(P_i(t)) = \cap_{i\in\mathcal{I}}\big(\mathcal{E}_i(P_i(t))\cap\Upsilon_M^{>}\big) = (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(t)))\cap\Upsilon_M^{>} \neq \emptyset.$$

Therefore, by (17), we know that $\forall t \in \overline{\{s\}} : \{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}(t) = 0$. However, since $\delta(s) \in \partial_K$, we know that there exists $sf \in \mathcal{L}(G)\setminus\mathcal{L}(H)$ such that $|f| = K$. Therefore, for the above sf, we have $(\forall vu \in \overline{\{sf\}} : vu \in \mathcal{L}(H) \wedge |u| \geq K)[\mathcal{D}_{i\in\mathcal{I}}^{\text{neg}}(v) = 0]$, i.e., (5) does not hold.

($\Leftarrow$) Suppose that the system is NSE-prognosable. First, we show that $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}$ satisfies (5). For any string $s \in \mathcal{L}(G)\setminus\mathcal{L}(H)$, since we assume that $d_{\min}(q_0) \geq K$ we know that there exists a nonfault prefix $s_N \in \overline{\{s\}}\cap\mathcal{L}(H)$ such that $\delta(s_N) \in \partial_K$. Furthermore, since H is NSE-prognosable, we know that there exists $t \in \overline{\{s_N\}}$ such that $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(t)))\cap\Upsilon_M^{>} = \emptyset$. By (16) and (17), we know that $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}(t) = 1$. Therefore, (5) is satisfied. Next, we show that $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}$ satisfies (6) by contradiction. Assume that (6) is not satisfied, i.e., there exist $s \in \mathcal{L}(H)$ and $t \in \mathcal{L}(H)/s$ such that $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}(s) = 1$ and $|t| \geq M$. Since $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}(s) = 1$, by (16) and (17), $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(s)))\cap\Upsilon_M^{>} = \emptyset$. Also, by $|t| \geq M$ and $st \in \mathcal{L}(H)$, we know that $d_{\max}(\delta(s)) \geq M$, i.e., $\delta(s) \in \Upsilon_M^{>}$. Note that, for any $i \in \mathcal{I}$, $\delta(s) \in \mathcal{E}_i(P_i(s))$. Therefore, $\delta(s) \in (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(s)))\cap\Upsilon_M^{>} \neq \emptyset$. This is a contradiction with the emptiness of $\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(s)))\cap\Upsilon_M^{>}$. Therefore, we know that (6) must be satisfied. ■

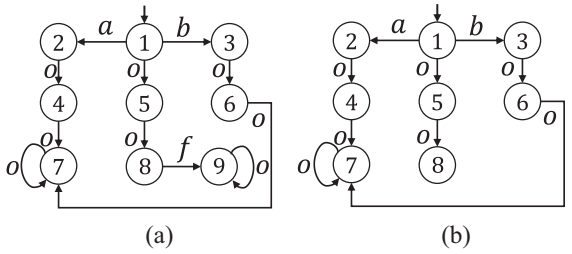


Fig. 2. H is NSE-prognosable with respect to G, $\Sigma_{o,1} = \{a, o\}$, $\Sigma_{o,2} = \{b, o\}$, $K = 0$, and $M = |Q_H|$. (a) G. (b) H.

The following example illustrates the negative protocol.

Example 2: Consider the system automaton G and the specification automaton H in Fig. 2. Suppose that there are two local agents 1 and 2, where $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. We consider performance bound $K = 0$ and $M = |Q_H| = 8$. Then we have $\partial_K = \{8\}$ and $\Upsilon_M^> = \{1, 2, 3, 4, 6, 7\}$. Since for the unique string leading to state 8, i.e., oo, there exists $o \in \overline{\{oo\}}$ such that $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(o))) \cap \Upsilon_M^> = \{5\} \cap \Upsilon_M^> = \emptyset$, we know that H is NSE-prognosable with respect to G and (M, K) and the negative protocol $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}$ achieves the two criteria in (5) and (6). To see this, let us still consider the unique string $oo \in \mathcal{L}(H)$ such that $\delta(oo) = 8 \in \partial_K$, where we have $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(oo))) \cap \Upsilon_M^> = \{7\} \neq \emptyset$. However, for its prefix $o \in \overline{\{oo\}}$, we have $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(o))) \cap \Upsilon_M^> = \emptyset$. Therefore, $\{\mathcal{D}_i^{\text{neg}}\}_{i\in\mathcal{I}}(o) = 1$, i.e., a fault alarm will be issued K steps before the fault occurs.

Remark 2: So far, we have developed two different decentralized prognositic protocols, i.e., the positive protocol and the negative protocol. One may ask which protocol is more powerful in the sense that the corresponding necessary and sufficient condition is weaker. In fact, these two protocols are *incomparable*, i.e., there may exist a system which is PSE-prognosable but is not NSE-prognosable; and vice versa. To see this, let us consider the system automaton G in Fig. 3(a), where we have $\mathcal{I} = \{1, 2\}$, $\Sigma_{o,1} = \{a, o\}$, and $\Sigma_{o,2} = \{b, o\}$. Let $K = 0$ and $M = 5$. For specification automaton H_{pos} shown in Fig. 3(b), we have $\partial_K = \{4\}$ and $\Upsilon_M^> = \{1, 2, 3, 5\}$ and for all strings leading to states 1, 2, 3, and 5, i.e., ϵ, a, b, ao, bo, aoo, boo, we have $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(\epsilon))) \cap \partial_K = (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(a))) \cap \partial_K = (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(b))) \cap \partial_K = (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(ao))) \cap \partial_K = (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(bo))) \cap \partial_K = (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(aoo))) \cap \partial_K = (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(boo))) \cap \partial_K = \emptyset$. Therefore, it is PSE-prognosable with respect to G. However, it is not NSE-prognosable, since for string $o \in \mathcal{L}(H_{\text{pos}}) : \delta(o) \in \partial_K$, we have $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(o))) \cap \Upsilon_M^> = \{3\} \neq \emptyset$ and for its prefix $\epsilon \in \mathcal{L}(H_{\text{pos}})$, we also have $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(\epsilon))) \cap \Upsilon_M^> = \{2\} \neq \emptyset$. On the other hand, if we consider the specification automaton H_{neg} shown in Fig. 3(c), then we have $\partial_K = \{3\}$ and $\Upsilon_M^> = \{1, 4, 5\}$ and it is NSE-prognosable with respect to G. However, it is not PSE-prognosable, since for string $o \in \mathcal{L}(H_{\text{neg}}) : \delta(o) \in \Upsilon_M^>$, we have $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(o))) \cap \partial_K = \{3\} \neq \emptyset$. Therefore, we conclude that the positive and the negative protocols are *incomparable* and we may need to apply different protocols to different systems.

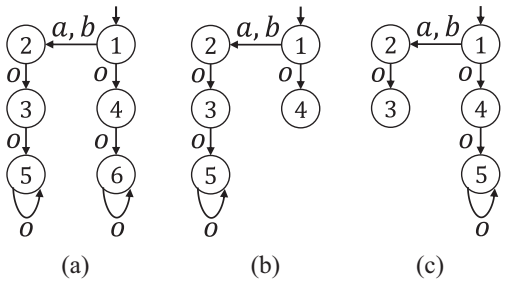


Fig. 3. Examples showing that PSE-prognosability and NSE-prognosability are incomparable. Let $K = 0, M = 5$, $\Sigma_{o,1} = \{a, o\}$, and $\Sigma_{o,2} = \{b, o\}$. Then H_{pos} is PSE-prognosable but not NSE-prognosable with respect to G, while H_{neg} is NSE-prognosable but not PSE-prognosable with respect to G. (a) G. (b) H_{pos}. (c) H_{neg}.

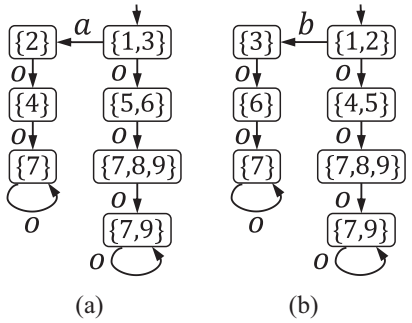


Fig. 4. Observer automata for system G shown in Fig. 2 where $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. (a) $\text{Obs}_1(G)$. (b) $\text{Obs}_2(G)$.

C. Implementations of State-Estimate-Based Protocols

We conclude this section by discussing the implementation issues of the state-estimate-based protocols. According to (12) and (16), in both the positive and the negative protocols, the key of implementing each local prognoser is to effectively compute the state-estimate $\mathcal{E}_i(P_i(s))$ for any observation $P_i(s)$. This can be done simply by constructing the standard *observer* automaton [3]. Specifically, for each $i \in \mathcal{I}$, its observer is

$$\text{Obs}_i(G) = \left(X_i^{\text{obs}}, \Sigma_{o,i}, f_i^{\text{obs}}, x_{0,i}^{\text{obs}}\right) \tag{19}$$

where $X_i^{\text{obs}} \subseteq 2^Q$ is the set of states, $\Sigma_{o,i}$ is the set of events, $f_i^{\text{obs}} : X_i^{\text{obs}} \times \Sigma_{o,i} \to X_i^{\text{obs}}$ is the transition function defined by: for any $x \in X_i^{\text{obs}}$ and $\sigma \in \Sigma_{o,i}$, we have

$$f_i^{\text{obs}}(x, \sigma) = \left\{q' \in Q : \exists q \in x, \exists w \in \left(\Sigma \setminus \Sigma_{o,i}\right)^* \text{ s.t. } q' = \delta(q, \sigma w)\right\}$$

and $x_{0,i}^{\text{obs}} \in X_i^{\text{obs}}$ is the initial state defined by

$$x_{0,i}^{\text{obs}} := \left\{q \in Q : \exists w \in \left(\Sigma \setminus \Sigma_{o,i}\right)^* \text{ s.t. } q = \delta(w)\right\}.$$

Note that the state space X_i^{obs} is defined *recursively* by $x_{0,i}^{\text{obs}}$ and f_i^{obs}, and hence, we only consider the reachable part of the observer automaton.

Example 3: Again, consider the system automaton G shown in Fig. 2, where $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. Then the observer automaton $\text{Obs}_1(G)$ for agent 1 is shown in Fig. 4(a). The initial state is $\{1, 3\}$ since event b could occur unobservably. Then, upon the occurrence of event o, we move to state $\{5, 6\}$ and so forth. Similarly, the observer automaton $\text{Obs}_2(G)$ for agent 2 is shown in Fig. 4(b).

Although computing the entire observer requires $2^{|Q|}$ states in the worst case, updating the current observer state can be done in polynomial-time since each observer state only contains at most Q states; see [3]. Therefore, for the purpose of online implementation, each local agent just need to store the current state-estimate $\mathcal{E}_i(P_i(s))$ and update it upon the occurrence of a new local observable event. The implementation of the coordinator simply requires to take the intersection of each state-estimate; the intersection operation can easily be implemented by either hardware or software. Moreover, the intersection operation at the coordinator is *memoryless* and *model-unaware*. In other words, most of the information are processed locally, which also meets the purpose of using decentralized architecture, and the fusion site can be simply implemented by simple memoryless devices.

IV. Verification of PSE-and NSE-Prognosability

In this section, we propose algorithms for the verifications of PSE-prognosability and NSE-prognosability, respectively. The verification algorithm for PSE-prognosability is based on the construction of the verifier automaton, while the verification algorithm for NSE-prognosability requires the construction of the observer automaton.

A. Verification of PSE-Prognosability

In order to verify PSE-prognosability, we use the verifier automaton that was introduced in the literature for the verification of codiagnosability [22], [26], [33]. Specifically, the verifier automaton is a finite-state automaton

$$V = \left(Q_V, \Sigma_V, \delta_V, q_{0,V}\right) \tag{20}$$

where

1) $Q_V \subseteq \underbrace{Q_H \times \cdots \times Q_H}_{(n+1)\text{ times}}$ is the set of states;
2) $\Sigma \subseteq \underbrace{(\Sigma \cup \{\epsilon\}) \times \cdots \times (\Sigma \cup \{\epsilon\})}_{(n+1)\text{ times}} \backslash \{(\epsilon, \ldots, \epsilon)\}$ is the set of events;
3) $q_{0,V} = (q_0, \ldots, q_0)$ is the initial state;
4) $\delta_V : Q_V \times \Sigma_V \to Q_V$ is the transition function defined as follows: for any $q_V = (q, q_1, \ldots, q_n)$ and $\sigma_V = (\sigma, \sigma_1, \ldots, \sigma_n)$, $\delta_V(q_V, \sigma_V)!$ if and only if the following conditions holds simultaneously:
   a) $\sigma \neq \epsilon \Rightarrow \delta_H(q, \sigma)!$
   b) $\forall i \in \mathcal{I} : \sigma_i \neq \epsilon \Rightarrow \delta_H(q_i, \sigma_i)!$
   c) $\forall i \in \mathcal{I} : P_i(\sigma) = P_i(\sigma_i)$.

   If $\delta_V(q_V, \sigma_V)!$, then we have

$$\delta_V(q_V, \sigma_V) = (\delta_H(q, \sigma), \delta_H(q_1, \sigma_1), \ldots, \delta_H(q_n, \sigma_n)). \tag{21}$$

Note that, each string in $\mathcal{L}(V)$ is a tuple and we write $s_V = (s, s_1, \ldots, s_n) \in \mathcal{L}(V)$, where $s, s_i \in \mathcal{L}(H)$. Note that, we only consider strings in $\mathcal{L}(H)$, since prognosability analysis does not consider what happens after the occurrence of fault. Intuitively, V tracks and only tracks all tuples $(s, s_1, \ldots, s_n)$, such that $\forall i \in \mathcal{I} : P_i(s) = P_i(s_i)$. That is, the first component represents the string in the real system and the $(i+1)$th component represents a string that looks the same as the string in the real system for agent i.

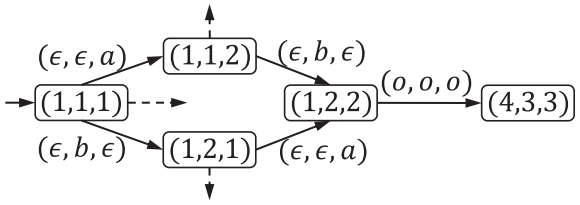


Fig. 5. Part of automaton V for system G and specification H_{neg} shown in Fig. 3.

The following theorem reveals how to use automaton V to verify PSE-prognosability.

Theorem 3: H is not PSE-prognosable with respect to G, $\Sigma_{o,i}$, $i \in \mathcal{I}$ and (M, K), if and only if, there exists a state $(q, q_1, \ldots, q_n) \in Q_V$ in automaton V such that

$$\left[q \in \Upsilon_M^{>}(G)\right] \wedge \left[q_1 = q_2 = \cdots = q_n \in \partial_K\right]. \tag{22}$$

Proof: ($\Rightarrow$) Suppose that H is not PSE-prognosable. Then we know that there exists $s \in \mathcal{L}(H)$ such that $\delta(s) \in \Upsilon_M^{>}(G)$ and $(\bigcap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_K \neq \emptyset$. Let $q \in (\bigcap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_K$. For each $i \in \mathcal{I}$, since $q \in \mathcal{E}_i(P_i(s))$, we know that there exists a string $s_i \in \mathcal{L}(H)$ such that $P_i(s) = P_i(s_i)$ and $q = \delta(s_i)$. By the definition of V, we know that there exists a string $s_V \in \mathcal{L}(V)$ such that its first component is s and its $(i+1)$th-component is s_i. Therefore, we know that state $(\delta(s), \delta(s_1), \ldots, \delta(s_n)) \in Q_V$ is reachable in V. Since $\delta(s) \in \Upsilon_M^{>}(G)$ and $q = \delta(s_1) = \cdots = \delta(s_n) \in \partial_K$, we know that the condition in (22) holds.

($\Leftarrow$) Suppose that there exists a state $q_V = (q, q_1, \ldots, q_n) \in Q_V$ in V such that (22) holds. Let $s_V \in \mathcal{L}(V)$ be a string in V such that $\delta_V(s_V) = q_V$. Note that string s_V is a tuple and we write $s_V = (s, s_1, \ldots, s_n)$. By the definition of V, we know that for each $i \in \mathcal{I}$, $P_i(s) = P_i(s_i)$. Therefore, we have $\{\delta(s), \delta(s_i)\} \subseteq \mathcal{E}_i(P_i(s))$. Moreover, since $q_1 = \cdots = q_n \in \partial_K$, we know that

$$\delta(s_1) = \cdots = \delta(s_n) \in (\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_K. \tag{23}$$

Therefore, we know that $\exists s \in \mathcal{L}(H)$ such that $\delta(s) \in \Upsilon_M^{>}$ and $(\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_K \neq \emptyset$, i.e., H is not PSE-prognosable. ■

We illustrate Theorem 3 by the following example.

Example 4: Let us consider again the system G shown in Fig. 3(a) and the specification H_{neg} shown in Fig. 3(c), where $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. We still consider $K = 0, M = 5$ and we have $\partial_K = \{3\}$ and $\Upsilon_M^{>} = \{1, 4, 5\}$. As discussed in Remark 2, H_{neg} is not PSE-prognosable with respect to G. Here, we show this by Theorem 3. Part of the verifier automaton V for G and H_{neg} is shown in Fig. 5. For reachable state $(q, q_1, q_2) = (4, 3, 3)$, we have $q = 4 \in \Upsilon_M^{>}$ and $q_1 = q_2 = 3 \in \partial_K$. Since state $(4, 3, 3)$ satisfies the condition in (22), we know that H_{neg} is not PSE-prognosable with respect to G.

Remark 3: Let us discuss the complexity of the above proposed verification procedure. As shown in [36], both sets ∂_K and $\Upsilon_M^{>}$ can be computed in $O(|\Sigma| \cdot |Q_H|^2)$. In the worst case, automaton V contains $|Q_H|^{n+1}$ states and

$(n+1)\cdot|\Sigma|\cdot|Q_H|^{n+1}$ transitions [22], [26], [33]. Moreover, determining whether or not V contains a state satisfying (22) is simply a reachability problem, which can be done linearly in the size of V. Therefore, the entire complexity for verifying PSE-prognosability using Theorem 3 is $O((n+1)\cdot|\Sigma|\cdot|Q_H|^{n+1})$.

B. Verification of NSE-Prognosability

Recall that, for each $i\in\mathcal{I}$, $\text{Obs}_i(G)$ is defined as the observer automaton with respect to $\Sigma_{o,i}$. Then we defined a new automaton $\tilde{G}$ by

$$\tilde{G}=\left(\tilde{Q},\Sigma,\tilde{\delta},\tilde{q}_0\right):=G\|\text{Obs}_1(G)\|\cdots\|\text{Obs}_n(G) \tag{24}$$

where "$\|$" denotes the usual *parallel composition* operation of automata; see [3, p. 80]. Then we have $\mathcal{L}(G)=\mathcal{L}(\tilde{G})$ and each state in $\tilde{G}$ is in the form of $\tilde{q}=(q,x_1,\ldots,x_n)$, where $q\in Q$ and $x_i\in 2^Q$. We call $\tilde{q}^1\xrightarrow{\sigma^1}\tilde{q}^2\xrightarrow{\sigma^2}\ldots\xrightarrow{\sigma^k}\tilde{q}^m$ a *path* in $\tilde{G}$ if $\tilde{q}^1$ is the initial state $\tilde{q}_0$ and $\forall k=1,\ldots,m-1:\tilde{\delta}(\tilde{q}^k,\tilde{\sigma}^k)=\tilde{q}^{k+1}$.

Next, we show how to use $\tilde{G}$ to verify NSE-prognosability.

Theorem 4: H is not NSE-prognosable with respect to G, $\Sigma_{o,i}$, $i\in\mathcal{I}$ and (M,K), if and only if, there exists a path

$$\left(q^0,x_1^0,\ldots,x_n^0\right)\xrightarrow{\sigma^1}\left(q^1,x_1^1,\ldots,x_n^1\right)\xrightarrow{\sigma^2}\cdots\xrightarrow{\sigma^m}\left(q^m,x_1^m,\ldots,x_n^m\right) \tag{25}$$

in $\tilde{G}$ such that:

1) $q^m\in\partial_K$;
2) $\forall k\in\{0,1\ldots,m\}:(\bigcap_{i\in\mathcal{I}}x_i^k)\cap\Upsilon_M^{>}\neq\emptyset$.

Proof: ($\Rightarrow$) Suppose that H is not NSE-prognosable, i.e., there exists a string $s\in\mathcal{L}(H)$ such that $\delta(s)\in\partial_K$ and $(\forall t\in\overline{\{s\}})[(\bigcap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(s)))\cap\Upsilon_M^{>}\neq\emptyset]$. Let $s=\sigma^1\ldots\sigma^{|s|}$, where $\sigma^i\in\Sigma$, and let

$$x^0\xrightarrow{\sigma^1}x^1\xrightarrow{\sigma^2}\ldots\xrightarrow{\sigma^{|s|}}x^{|s|}$$

be the path induced by s from $x^0=\tilde{x}_0$. For each $k\in\{0,1,\ldots,m\}$, we write $x^k=(q^k,x_1^k,\ldots,x_n^k)$. We claim that this path satisfies the two conditions in the theorem. First, since $q^{|s|}=\delta(s)\in\partial_K$, we know that the first condition holds. Second, for any $k\in\{0,1,\ldots,|s|\}$ and $i\in\mathcal{I}$, by the property of observer, we have that

$$x_i^k=f_i^{\text{obs}}\left(P_i\left(\sigma^1\ldots\sigma^k\right)\right)=\mathcal{E}_i\left(P_i\left(\sigma^1\ldots\sigma^k\right)\right).$$

Since $\sigma^1\ldots\sigma^k\in\overline{\{s\}}$, we know that $(\bigcap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(\sigma^1\ldots\sigma^k)))\cap\Upsilon_M^{>}\neq\emptyset$. Therefore, $(\bigcap_{i\in\mathcal{I}}x_i^k)\cap\Upsilon_M^{>}\neq\emptyset$ for any $k\in\{0,1,\ldots,|s|\}$, i.e., the second condition holds.

($\Leftarrow$) Suppose that there exists a path in (25) such that the two conditions in the theorem hold. Let us consider string $s=\sigma^1\ldots\sigma^m$, where $\sigma^i\in\Sigma$ are events in (25). Still, we know that $\delta(s)=q^m\in\partial_K$. Also, for any $t\in\overline{\{s\}}$, it can be written by $t=\sigma^1\ldots\sigma^{|t|}$. Since

$$\mathcal{E}_i(P_i(t))=f_i^{\text{obs}}\left(P_i\left(\sigma^1\ldots\sigma^{|t|}\right)\right)=x_i^{|t|}$$

we have

$$\left(\bigcap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(t))\right)\cap\Upsilon_M^{>}=\left(\bigcap_{i\in\mathcal{I}}x_i^{|t|}\right)\cap\Upsilon_M^{>}\neq\emptyset. \tag{26}$$

This implies that H is not NSE-prognosable. ■

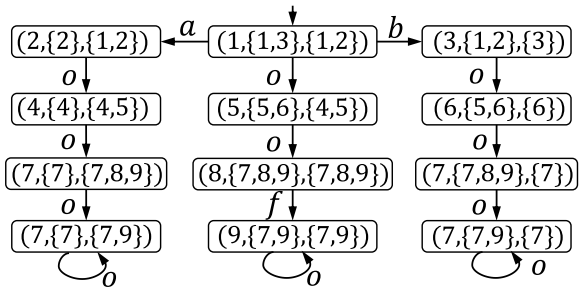


Fig. 6. Automaton $\tilde{G}$ for system G and specification H shown in Fig. 2.

We illustrate Theorem 4 by the following example.

Example 5: Let us consider again the system G and the specification H shown in Fig. 2, where $\Sigma_{o,1}=\{a,o\}$ and $\Sigma_{o,2}=\{b,o\}$. We still consider performance bound $K=0$ and $M=5$; we have $\partial_K=\{8\}$ and $\Upsilon_M^{>}=\{1,2,3,4,6,7\}$. We have discussed in Example 2 that H is NSE-prognosable with respect to G. Here, we show this by Theorem 4. First, we construct automaton $\tilde{G}$, which is shown in Fig. 6. Note that the only string that leads to a state, whose first component is in ∂_K, is oo. However, for the second state in path $(1,\{1,3\},\{1,2\})\xrightarrow{o}(5,\{5,6\},\{4,5\})\xrightarrow{o}(8,\{7,8,9\},\{7,8,9\})$, we have $\{5,6\}\cap\{4,5\}\cap\Upsilon_M^{>}=\{5\}\cap\Upsilon_M^{>}=\emptyset$. Therefore, there does not exist a path in $\tilde{G}$ satisfying the two conditions in Theorem 4 and we know that H is NSE-prognosable.

Remark 4: Let us explain why we propose the observer-based approach to verify NSE-prognosability instead of using the verifier-based approach. Note that PSE-prognosability is a pure *always-type* property that should be satisfied for all strings leading to $\Upsilon_M^{>}$, while for NSE-prognosability, we need to check the satisfaction of the condition for *some* prefix of any string leading to ∂_K. However, automaton V only tracks state pairs that cannot be distinguished by each agent. For an *always-type* property, we can look at each pair, since one pair that violates the condition will lead to the violation of the entire condition. However, to determine the existence of some prefix that satisfies the condition, we cannot make such a conclusion as the violation of the condition does not necessary imply that it does not has a prefix that can "save" it. Therefore, we need to use the observer automaton to track how each state is visited in order to determine whether or not it has a prefix that can save it.

Remark 5: We conclude this section by analyzing the complexity of the above proposed procedure for the verification of NSE-prognosability. For states in $\tilde{G}$, we define

$$X_A:=\left\{(q,x_1,\ldots,x_n)\in\tilde{X}:(\cap_{i\in\mathcal{I}}x_i)\cap\Upsilon_M^{>}\neq\emptyset\right\}$$
$$X_B:=\left\{(q,x_1,\ldots,x_n)\in\tilde{X}:q\in\partial_K\right\}.$$

Therefore, to check the condition in Theorem 4, it suffices to check whether there exists a state in $X_A\cap X_B$ that can be reached from the initial state only via states in X_A. This problem equals to whether or not a state in X_B can be reached if we remove all states in $\tilde{X}\setminus X_A$, which is simply a reachability problem that can be solved linearly with respect to the

number of transitions in $\tilde{G}$ via a depth-first search. Note that automaton $\tilde{G}$ contains at most $|Q| \cdot 2^{n \cdot |Q|}$ states. Therefore, the entire complexity for checking NSE-prognosability using the proposed approach is $O(|\Sigma| \cdot |Q| \cdot 2^{n \cdot |Q|})$.

V. Complexity of the Verification Problem

In the above section, procedures for the verifications of PSE-prognosability and NSE-prognosability have been proposed, respectively. The verification of NSE-prognosability is exponential in the size of G, while verification of PSE-prognosability is polynomial in the size of G. However, both procedures are exponential with respect to the number of local agents, i.e., n. In this section, we show that the verifications of PSE-prognosability and NSE-prognosability are both PSPACE-hard with respect to n. Therefore, this exponential complexity as the number of agents increases seems to be unavoidable.

To show a decision problem is PSPACE-hard, we need to reduce it to a known PSPACE-hard/complete problem in polynomial-time. One well-known PSPACE-hard problem is the deterministic finite-state automata intersection problem DFA intersection (DFA-Int) [17] stated as follows.

DFA Intersection Problem.

1) *Instance:* A set of automata $\{G_1, G_2, \ldots, G_n\}$.
2) *Question:* Whether or not $\cap_{i\in\{1,\ldots,n\}} \mathcal{L}_m(G_i) = \emptyset$.

Hereafter, we will use DFA-Int to show the PSPACE-hardness of PSE-prognosability and NSE-prognosability.

First, we show that the verification of PSE-prognosability is PSPACE-hard.

Theorem 5: The verification of PSE-prognosability is PSPACE-hard with respect to the number of local agents.

Proof: In order to prove this result, we reduce DFA-Int to the PSE-prognosability verification problem. Let $\{G_1, G_2, \ldots, G_n\}$ be the instance of DFA-Int, where $G_i = (Q_i, \Sigma, \delta_i, q_{0,i}, Q_{m,i})$.[1] Then we construct a new automaton

$$G_{\text{red}} = \left(Q_{\text{red}}, \Sigma_{\text{red}}, \delta_{\text{red}}, q_{0,\text{red}}\right) \tag{27}$$

where

1) $Q_{\text{red}} = Q_1 \dot{\cup} \ldots \dot{\cup} Q_n \dot{\cup} \{q_{0,\text{red}}, A, B, C, D, E\}$ is the set of states;
2) $\Sigma_{\text{red}} = \Sigma \dot{\cup} \{f_1, f_2, \sigma, e, e_1, \ldots, e_n\}$ is the set of events;
3) $q_{0,\text{red}}$ is a new initial state;
4) The transition function $\delta_{\text{red}} : Q_{\text{red}} \times \Sigma_{\text{red}} \to Q_{\text{red}}$ is defined as follows: a) for each $i \in \{1, \ldots, n\}$, the transitions between states in Q_i are consistent with G_i; b) for each $i \in \{1, \ldots, n\}$, $\delta_{\text{red}}(q_{0,\text{red}}, e_i) = q_{0,i}$ is defined; c) for each $q \in Q_{m,1} \cup \cdots \cup Q_{m,n}$, $\delta_{\text{red}}(q, \sigma) = A$ is defined; and d) the remaining transitions to states B, C, D, E are specified by Fig. 7.

Based on system automaton G_{red}, we define a specification automaton H_{red} by removing state B and its associated transitions. Let $K = 0$ and $M = 1$. Then we have $\partial_0 = \{A\}$ and $\Upsilon_1^{>} = Q_1 \cup \cdots \cup Q_n \cup \{q_{0,\text{red}}, C, D, E\}$. Also, we consider n local agents; each of them can observe events $\Sigma_{o,i} = \Sigma \cup \{\sigma, e_1, \ldots, e_n\} \setminus \{e_i\}$. Clearly, constructing G_{red} and H_{red} can be done in polynomial-time with respect to the instance of DFA-Int.

[1] We assume w.l.o.g. that all automata have the same event set Σ.

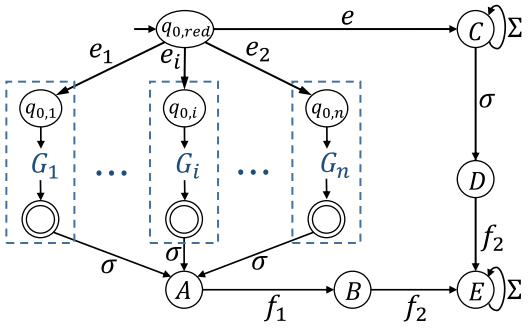


Fig. 7. Conceptual illustration of how to construct G_{red} from $\{G_1, \ldots, G_n\}$.

Hereafter, we show that H_{red} is PSE-prognosable with respect to G_{red}, $\Sigma_{o,i}, i \in \mathcal{I}$ and (M, K) if and only if $\cap_{i\in\mathcal{I}} \mathcal{L}_m(G_i) = \emptyset$.

($\Rightarrow$) By contraposition. Suppose that $\cap_{i\in\mathcal{I}} \mathcal{L}_m(G_i) \neq \emptyset$ and let $s \in \cap_{i\in\mathcal{I}} \mathcal{L}_m(G_i)$. By the construction of H_{red}, we know that for each $i \in \mathcal{I}$, $e_i s\sigma \in \mathcal{L}(H_{\text{red}})$ and $\delta_{\text{red}}(e_i s\sigma) = A \in \partial_0$. Also, we have that $es\sigma \in \mathcal{L}(H_{\text{red}})$ and $\delta_{\text{red}}(es\sigma) = D \in \Upsilon_1^{>}$. Since $P_i(e_i s\sigma) = P_i(es\sigma)$, we know that $\{A, D\} \subseteq \mathcal{E}_i(P_i(es\sigma))$, which further implies that $A \in (\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(es\sigma))) \cap \partial_0 \neq \emptyset$. Therefore, H_{red} is not PSE-prognosable with respect to G_{red}.

($\Leftarrow$) Still by contraposition. Suppose that H_{red} is not PSE-prognosable with respect to G_{red}, i.e.,

$$\left(\exists s \in \mathcal{L}(H_{\text{red}}) : \delta_{\text{red}}(s) \in \Upsilon_1^{>}\right)\left[(\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_0 \neq \emptyset\right].$$

For the above s that violates PSE-prognosability, it must contain a prefix in the form of $s' = ew\sigma$; if s does not contain σ, then each agent knows for sure that the system is not in A, which is the unique state in ∂_0, i.e., $(\cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(s))) \cap \partial_0 = \emptyset$. Therefore, we know that $A \in \cap_{i\in\mathcal{I}} \mathcal{E}_i(P_i(ew\sigma))$, i.e., for each agent $i \in \mathcal{I}$, there exists $s_i \in \mathcal{L}(H_{\text{red}})$ such that $P_i(ew\sigma) = P_i(s_i)$ and $\delta_{\text{red}}(s_i) = A$. By the construction of G_{red}, s_i can only be $e_i w\sigma$. Therefore, $\forall i \in \mathcal{I} : e_i w\sigma \in \mathcal{L}(G_{\text{red}})$, which further implies that $\forall i \in \mathcal{I} : w \in \mathcal{L}_m(G_i)$. Therefore, $w \in \cap_{i\in\mathcal{I}} \mathcal{L}_m(G_i) \neq \emptyset$. ■

Next, we show that the verification of NSE-prognosability is also PSPACE-hard.

Theorem 6: The verification of NSE-prognosability is PSPACE-hard with respect to the number of local agents.

Proof: In order to prove this result, we still reduce DFA-Int to the NSE-prognosability verification problem. (Another possible way is to reduce the PSE-prognosability verification problem to the NSE-prognosability verification problem.) Let $\{G_1, G_2, \ldots, G_n\}$ the instance of DFA-Int, where $G_i = (Q_i, \Sigma, \delta_i, q_{0,i}, Q_{m,i})$. We construct a new system automaton $G_{\text{red}} = (Q_{\text{red}}, \Sigma_{\text{red}}, \delta_{\text{red}}, q_{0,\text{red}})$ following the same construction in the proof of Theorem 5. However, the specification automaton H_{red} is constructed by removing state E and its associated transitions. Let $K = 0$ and $M = 1$. Then we have $\partial_0 = \{B, D\}$ and $\Upsilon_1^{>} = Q_1 \cup \cdots \cup Q_n \cup \{q_{0,\text{red}}, A, C\}$. Still, we consider n local agents; each of them can observe events $\Sigma_{o,i} = \Sigma \cup \{\sigma, f_1, e_1, \ldots, e_n\} \setminus \{e_i\}$. Clearly, constructing G_{red}

and H_{red} can still be done in polynomial-time with respect to the instance of DFA-Int.

Hereafter, we show that H_{red} is NSE-prognosable with respect to G_{red}, $\Sigma_{o,i}, i \in \mathcal{I}$ and (M, K) if and only if $\cap_{i\in\mathcal{I}}\mathcal{L}_m(G_i) = \emptyset$.

($\Rightarrow$) By contrapositive. Suppose that $\cap_{i\in\mathcal{I}}\mathcal{L}_m(G_i) \neq \emptyset$ and let $s \in \cap_{i\in\mathcal{I}}\mathcal{L}_m(G_i)$. Then we know that for each $i \in \mathcal{I}$, $e_i s\sigma \in \mathcal{L}(H_{\text{red}})$ and $\delta_{\text{red}}(e_i s\sigma) = A \in \Upsilon_1^>$. Since $P_i(e_i s\sigma) = P_i(es\sigma)$ we know that $\{A, D\} \subseteq \mathcal{E}_i(P_i(es\sigma))$. Note that $\delta_{\text{red}}(es\sigma) = D \in \partial_0$. Then for each string in

$$\overline{\{es\sigma\}} = \left\{\epsilon, e, e\sigma^1, e\sigma^1\sigma^2, \ldots, e\sigma^1\sigma^2\cdots\sigma^{|s|}, es\sigma\right\} \quad (28)$$

where $s = \sigma^1\sigma^2\ldots\sigma^{|s|}$, we have

$$\begin{aligned} q_{0,\text{red}} &\in \cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(\epsilon)) \cap \Upsilon_1^>(G) \\ &= \cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(e)) \cap \Upsilon_1^>(G) \neq \emptyset \\ C &\in \left(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(e\sigma^1\cdots\sigma^i))\right) \cap \Upsilon_1^>(G) \neq \emptyset \\ &\quad i = 1, \ldots, |s| \\ A &\in (\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(es\sigma))) \cap \Upsilon_1^>(G) \neq \emptyset. \end{aligned}$$

Therefore, H_{red} is not NSE-prognosable with respect to G_{red}.

($\Leftarrow$) Since $\partial_0 = \{B, D\}$, any string leading to ∂_0 must be either in the form of $e_i s\sigma f_1$, where $s \in \mathcal{L}_m(G_i)$, or in the form of $es\sigma$, where $s \in \Sigma^*$. For any string in the form of $e_i s\sigma f_1$, since the only state that can be reached in G_{red} after observing f_1 is B, we know that $\mathcal{E}_i(P_i(e_i s\sigma f_1)) = \{B, E\}$. Since $B \notin \Upsilon_1^>$, we have immediately that $(\bigcap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(e_i s\sigma f_1))) \cap \Upsilon_1^> = \emptyset$. For string in the form of $es\sigma$, we know that there must exists $k \in \mathcal{I}$ such that $e_k s\sigma \notin \mathcal{L}(G_{\text{red}})$; otherwise, we know that $s \in \cap_{i\in\mathcal{I}}\mathcal{L}_m(G_i)$, which violates the assumption that $\cap_{i\in\mathcal{I}}\mathcal{L}_m(G_i) = \emptyset$. Therefore, for the above $k \in \mathcal{I}$, we have $\mathcal{E}_k(P_k(e_k s\sigma)) = \{D\}$. Since $D \notin \Upsilon_1^>$, we know that $(\bigcap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(es\sigma\sigma))) \cap \Upsilon_1^> = \emptyset$. Overall, we know that

$$(\forall s \in \mathcal{L}(H_{\text{red}}) : \delta_{\text{red}}(s) \in \partial_0)\left[\left(\bigcap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(s))\right) \cap \Upsilon_1^> = \emptyset\right]$$

that is H_{red} is NSE-coprognosable with respect to G_{red}. ■

VI. Comparison With Other Decentralized Protocol

In this section, we compare the proposed two protocols with existing decentralized prognostic protocols in the literature. Specifically, we compare the positive and negative protocols with the inference-based protocol [29] and the conjunctive protocol [16]. We do not consider the disjunctive protocol [18] since it is subsumed by the inference-based protocol [29]. Hereafter, we show that each of the proposed protocols is *incomparable* with any of the existing protocols.

First, we recall the notion of N-inference prognosability from [29], which is the necessary and sufficient condition under which the N-inference-based protocol satisfies (5) and (6) with $K = 0$ and $M = |Q_H|$. First, we define

$$\begin{aligned} \partial_{\mathcal{L}} &\coloneqq \{s \in \mathcal{L}(H) : \delta(s) \in \partial_0\} \\ \Upsilon_{\mathcal{L}} &\coloneqq \left\{s \in \mathcal{L}(H) : \delta(s) \in \Upsilon_{|Q|}^>\right\} \\ \Im_{\mathcal{L}} &\coloneqq \mathcal{L}(H) \setminus \Upsilon_{\mathcal{L}}. \end{aligned}$$

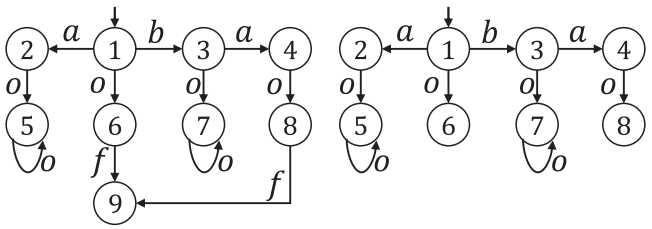


Fig. 8. System is both PSE-prognosable and NSE-prognosable, but it is neither N-inference prognosable nor conjunctively prognosable, where $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. (a) G. (b) H.

Now, we are ready to recall N-inference prognosability.

Definition 3: Specification H is said to be N-inference prognosable with respect to G and $\Sigma_{o,i}, i \in \mathcal{I}$ if

$$\partial_{\mathcal{L}} \subseteq \Im_{\mathcal{L}} \setminus \Im_{\mathcal{L}}[N+1] \quad (29)$$

where $\Im_{\mathcal{L}}[N+1]$ is defined inductively as follows.

1) $\Im_{\mathcal{L}}[0] \coloneqq \Im_{\mathcal{L}}$ and $\Upsilon_{\mathcal{L}}[0] = \Upsilon_{\mathcal{L}}$.
2) For any $k \geq 0$, we have

$$\begin{cases} \Im_{\mathcal{L}}[k+1] \coloneqq \Im_{\mathcal{L}}[k] \cap \left(\bigcap_{i\in\mathcal{I}} P_i^{-1}(P_i(\Upsilon_{\mathcal{L}}[k]))\right) \\ \Upsilon_{\mathcal{L}}[k+1] \coloneqq \Upsilon_{\mathcal{L}}[k] \cap \left(\bigcap_{i\in\mathcal{I}} P_i^{-1}(P_i(\Im_{\mathcal{L}}[k]))\right). \end{cases} \quad (30)$$

Also, we recall the notion of conjunctive coprognosability from [16], which is the necessary and sufficient condition under which the conjunctive protocol satisfies (5) and (6) with $K = 0$ and $M = |Q_H|$.

Definition 4: H is said to be conjunctively coprognosable with respect to G and $\Sigma_{o,i}, i \in \mathcal{I}$ if, $\cap_{i\in\mathcal{I}}[P_i^{-1}P_i(\partial_{\mathcal{L}})] \cap \Upsilon_{\mathcal{L}} = \emptyset$.

Remark 6: Note that, in conjunctive prognosability and N-inference prognosability, the performance bound is considered implicitly as $K = 0$ and $M = |Q_H|$. Therefore, to compare PSE-prognosability and NSE-prognosability with these two notions, hereafter, we will also fix the performance bound as $K = 0$ and $M = |Q_H|$.

The following example shows that the positive and the negative protocols may predict the fault correctly when the existing protocols fail.

Example 6: Let us consider the system G shown in Fig. 8(a) and the specification H shown in Fig. 8(b), where $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. We have $\partial_{\mathcal{L}} = \{o, bao\}$, $\Im_{\mathcal{L}} = \{o, ba, bao\}$, and $\Upsilon_{\mathcal{L}} = \{\epsilon\} \cup \{a, b\}\{o\}^*$. Then we have

$$\begin{aligned} &\Im_{\mathcal{L}}[0] = \{o, ba, bao\}, \Upsilon_{\mathcal{L}}[0] = \{\epsilon, a, b\} \cup \{ao, bo\}\{o\}^* \\ &\Im_{\mathcal{L}}[1] = \{o, ba, bao\}, \Upsilon_{\mathcal{L}}[1] = \{ao, bo\} \\ &\Im_{\mathcal{L}}[k] = \{o, bao\}, \Upsilon_{\mathcal{L}}[k] = \{ao, bo\}, \forall k \geq 2. \end{aligned}$$

Therefore, $\partial_{\mathcal{L}} \not\subseteq \Im_{\mathcal{L}} \setminus \Im_{\mathcal{L}}[N+1]$ for any $N \geq 0$, i.e., the system is not N-inference prognosable for any N and the fault cannot be correctly predicted by using the inference-based protocol. Also, we have

$$\cap_{i\in\mathcal{I}}\left[P_i^{-1}P_i(\partial_{\mathcal{L}})\right] \cap \Upsilon_{\mathcal{L}} = \{ao, bo\} \neq \emptyset.$$

Therefore, the system is not conjunctively coprognosable, i.e., the fault still cannot be correctly predicted by using

the conjunctive protocol. However, one can verify, by the proposed algorithms, that H is both PSE-prognosable and NSE-prognosable for $K = 0$ and $M = |Q_H|$. Therefore, either of the propose protocols can correctly predict the fault when all existing protocols fail.

On the other hand, the proposed protocols may fail to predict fault correctly when the existing protocols can do so. This is illustrated by the following example.

Example 7: As shown in Remark 2, specification H_{pos} in Fig. 3(b) is not NSE-prognosable with respect to G in Fig. 3(a). However, we have $\partial_{\mathcal{L}} = \{o\}$, $\Upsilon_{\mathcal{L}} = \{\epsilon\} \cup \{a, b\}\{o\}^*$ and

$$\cap_{i\in\mathcal{I}}\left[P_i^{-1}P_i(\partial_{\mathcal{L}})\right] \cap \Upsilon_{\mathcal{L}} = \{o, ao\} \cap \{o, bo\} \cap \Upsilon_{\mathcal{L}} = \emptyset.$$

Therefore, H_{pos} is conjunctively coprognosable with respect to G.

Also, as shown in Remark 2, specification H_{neg} in Fig. 3(c) is not PSE-prognosable with respect to G in Fig. 3(a). However, we have $\partial_{\mathcal{L}} = \{ao, bo\}$, $\Im_{\mathcal{L}} = \{a, b, ao, bo\}$, $\Upsilon_{\mathcal{L}} = \{o\}^*$ and

$$\Im_{\mathcal{L}}[1] = \Im_{\mathcal{L}} \cap \left(\cap_{i\in\mathcal{I}} P_i^{-1}(P_i(\Upsilon_{\mathcal{L}}))\right) = \emptyset$$
$$\partial_{\mathcal{L}} = \{ao, bo\} \subseteq \Im_{\mathcal{L}} \setminus \Im_{\mathcal{L}}[1] = \{a, b, ao, bo\}.$$

Therefore, H_{neg} is 0-inference prognosable with respect to G.

Recall that we have shown in Section III that PSE-prognosability and NSE-prognosability are incomparable. Moreover, as shown in [16], conjunctive coprognosability and N-inference prognosability are also incomparable. Therefore, we conclude that all of these four notions are incomparable in general; each of them may apply to different systems.

Finally, it is worth remarking that, in this paper, we adopt a *state-based* approach, while the results in [16], [18], and [29] use *language-based* frameworks. As a consequence, the proposed protocols strictly depend on the state-space of the system that generates the language. In other words, it is possible that we can refine the state-space of a non-PSE-(or NSE-) prognosable system such that the refined system is PSE-(or NSE-) prognosable. This issue is illustrated by the following example.

Example 8: Let us consider the system automaton G' in Fig. 9(a), where we have $\mathcal{I} = \{1, 2\}$, $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. Let $K = 0$ and $M = 5$. For specification automaton H'_{pos} shown in Fig. 9(b), we have $\partial_K = \{4\}$ and $\Upsilon_M^{>} = \{1, 2, 3, 5, 7, 8, 9\}$. It is NSE-prognosable, since for string $o \in \mathcal{L}(H'_{\text{pos}}) : \delta(o) \in \partial_K$, we have $(\cap_{i\in\mathcal{I}}\mathcal{E}_i(P_i(o))) \cap \Upsilon_M^{>} = \emptyset$. However, as we have shown in Remark 2, that H_{pos} in Fig. 3(b) is not NSE-prognosable with respect to G_{pos} in Fig. 3(a) under the same problem parameters. Interestingly, we see that $\mathcal{L}(G') = \mathcal{L}(G)$ and $\mathcal{L}(H'_{\text{pos}}) = \mathcal{L}(H_{\text{pos}})$, i.e., G' and $H_{\text{pos}'}$ are just refinements of G and H_{pos}, respectively. This justifies our early assertion that, even the languages of the automata are the same, using different state-spaces of the system may affect the results of the protocols. Intuitively, this can be explained as follows. By splitting states of the system, each state can carry more information. However, as a tradeoff, more communication capacity is needed between each local site and the coordinator.

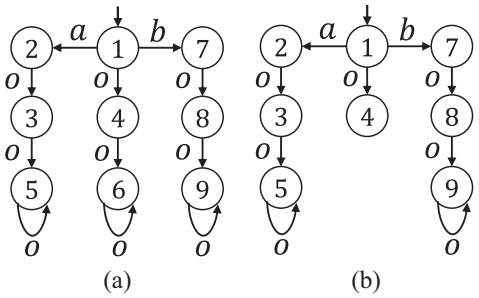


Fig. 9. H'_{pos} is NSE-prognosable with respect to G', where $\Sigma_{o,1} = \{a, o\}$, $\Sigma_{o,2} = \{b, o\}$, $K = 0$, and $M = 5$. (a) G'. (b) H'_{pos}.

VII. Conclusion

In this paper, two novel decentralized protocols were proposed for the purpose of fault prognosis. The notions of PSE-prognosability and NSE-prognosability were introduced, respectively, as the necessary and sufficient conditions under which the positive protocol and negative protocol achieve the required performance bound, respectively. Algorithms were provided to verify the proposed notions. We showed that the verifications of PSE-prognosability and NSE-prognosability are both PSPACE-hard with respect to the number of local agents. Finally, we showed that the two proposed protocols are incomparable with existing protocols in the literature.

The proposed algorithms are exponential in the number of local agents, which has been shown to be unavoidable. Note that the verification algorithm for PSE-prognosability requires polynomial-time with respect to the size of the system model, while the algorithm for the verification NSE-prognosability requires exponential-time with respect to the size of the system model. Investigating whether or not there exists a polynomial-time algorithm for the verification of NSE-prognosability is an interesting future direction.

References

[1] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre, "Fault prognosis of timed stochastic discrete event systems with bounded estimation error," *Automatica*, vol. 82, pp. 35–41, Aug. 2017.

[2] N. Bertrand, S. Haddad, and E. Lefaucheux, "Foundation of diagnosis and predictability in probabilistic systems," in *Proc. 34th IARCS Annu. Conf. FSTTCS*, 2014, pp. 417–429.

[3] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York, NY, USA: Springer, 2008.

[4] F. Cassez and A. Grastien, "Predictability of event occurrences in timed systems," in *Formal Modeling and Analysis of Timed Systems*. Heidelberg, Germany: Springer, 2013, pp. 62–76.

[5] M. Chang, W. Dong, Y. Ji, and L. Tong, "On fault predictability in stochastic discrete event systems," *Asian J. Control*, vol. 15, no. 5, pp. 1458–1467, 2013.

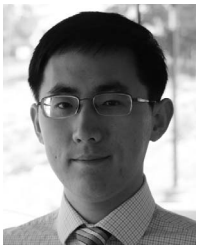
[6] J. Chen and R. Kumar, "Failure detection framework for stochastic discrete event systems with guaranteed error bounds," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1542–1553, Jun. 2015.

[7] J. Chen and R. Kumar, "Stochastic failure prognosability of discrete event systems," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1570–1581, Jun. 2015.

[8] H. Cho and S. I. Marcus, "On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation," *Math. Control Signals Syst.*, vol. 2, no. 1, pp. 47–69, 1989.

[9] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discr. Event Dyn. Syst. Theory Appl.*, vol. 10, nos. 1–2, pp. 33–86, 2000.

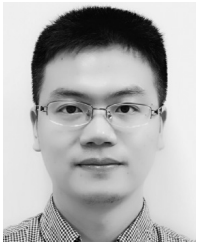
[10] S. Genc and S. Lafortune, "Predictability of event occurrences in partially-observed discrete-event systems," *Automatica*, vol. 45, no. 2, pp. 301–311, 2009.
[11] T. Jéron, H. Marchand, S. Genc, and S. Lafortune, "Predictability of sequence patterns in discrete event systems," in *Proc. 17th IFAC World Congr.*, Seoul, South Korea, 2008, pp. 537–543.
[12] C. Keroglou and C. N. Hadjicostis, "Distributed diagnosis using predetermined synchronization strategies in the presence of communication constraints," in *Proc. IEEE CASE*, 2015, pp. 831–836.
[13] A. Khoumsi and H. Chakib, "Decentralized supervisory control of discrete event systems: Involving the fusion system in the decision-making," in *Proc. Int. Conf. Intell. Syst. Control (IASTED)*, Cambridge, MA, USA, 2007, pp. 44–49.
[14] A. Khoumsi and H. Chakib, "A new architecture for decentralized control of discrete event systems: Decidability and synthesis issues," in *Proc. Conf. Francophone de Modélisation et Simulat.*, Paris, France, 2008.
[15] A. Khoumsi and H. Chakib, "Multi-decision decentralized prognosis of failures in discrete event systems," in *Proc. Amer. Control Conf.*, St. Louis, MO, USA, 2009, pp. 4974–4981.
[16] A. Khoumsi and H. Chakib, "Conjunctive and disjunctive architectures for decentralized prognosis of failures in discrete-event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 412–417, Apr. 2012.
[17] D. Kozen, "Lower bounds for natural proof systems," in *Proc. 18th Symp. Found. Comput. Sci.*, Providence, RI, USA, 1977, pp. 254–266.
[18] R. Kumar and S. Takai, "Decentralized prognosis of failures in discrete event systems," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 48–59, Jan. 2010.
[19] R. H. Kwong and D. L. Yonge-Mallo, "Fault diagnosis in discrete-event systems: Incomplete models and learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 118–130, Feb. 2011.
[20] R. H. Kwong and D. L. Yonge-Mallo, "Fault diagnosis in discrete-event systems with incomplete models: Learnability and diagnosability," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1236–1249, Jul. 2015.
[21] D. Lefebvre, "Fault diagnosis and prognosis with partially observed Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 10, pp. 1413–1424, Oct. 2014.
[22] M. V. Moreira, T. C. Jesus, and J. C. Basilio, "Polynomial time verification of decentralized diagnosability of discrete event systems," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1679–1684, Jul. 2011.
[23] F. Nouioua, P. Dague, and L. Ye, "Probabilistic analysis of predictability in discrete event systems," in *Proc. 25th Int. Workshop Principles Diagnosis*, 2014.
[24] F. Nouioua, P. Dague, and L. Ye, "Predictability in probabilistic discrete event systems," in *Soft Methods for Data Science*. Cham, Switzerland: Springer, 2017, pp. 381–389.
[25] M. Panteli and C. N. Hadjicostis, "Intersection based decentralized diagnosis: Implementation and verification," in *Proc. 52nd IEEE Conf. Decis. Control*, Florence, Italy, 2013, pp. 6311–6316.
[26] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 2, pp. 384–395, Mar. 2006.
[27] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.
[28] S. Takai, "Robust prognosability for a set of partially observed discrete event systems," *Automatica*, vol. 51, pp. 123–130, Jan. 2015.
[29] S. Takai and R. Kumar, "Inference-based decentralized prognosis in discrete event systems," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 165–171, Jan. 2011.
[30] S. Takai and R. Kumar, "Distributed failure prognosis of discrete event systems with bounded-delay communications," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1259–1265, May 2012.
[31] L. Ye, P. Dague, and F. Nouioua, "Predictability analysis of distributed discrete event systems," in *Proc. 52nd Conf. Decis. Control*, Florence, Italy, 2013, pp. 5009–5015.
[32] X. Yin, "Verification of prognosability for labeled Petri nets," *IEEE Trans. Autom. Control*, to be published, doi: 10.1109/TAC.2017.2756096.
[33] X. Yin and S. Lafortune, "Codiagnosability and coobservability under dynamic observations: Transformation and verification," *Automatica*, vol. 61, pp. 241–252, Nov. 2015.
[34] X. Yin and S. Lafortune, "A general approach for solving dynamic sensor activation problems for a class of properties," in *Proc. 54th IEEE Conf. Decis. Control*, Osaka, Japan, 2015, pp. 3610–3615.
[35] X. Yin and S. Lafortune, "Decentralized supervisory control with intersection-based architecture," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3644–3650, Nov. 2016.
[36] X. Yin and Z.-J. Li, "Decentralized fault prognosis of discrete event systems with guaranteed performance bound," *Automatica*, vol. 69, pp. 375–379, Jul. 2016.
[37] X. Yin and Z.-J. Li, "Reliable decentralized fault prognosis of discrete-event systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1598–1603, Nov. 2016.
[38] S. Yokota, T. Yamamoto, and S. Takai, "Computation of the delay bounds and synthesis of diagnosers for decentralized diagnosis with conditional decisions," *Discr. Event Dyn. Syst. Theory Appl.*, vol. 27, no. 1, pp. 45–84, 2017.
[39] M. Yokotani, T. Kondo, and S. Takai, "Abstraction-based verification and synthesis for prognosis of discrete event systems," *Asian J. Control*, vol. 18, no. 4, pp. 1279–1288, 2016.
[40] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annu. Rev. Control*, vol. 37, no. 2, pp. 308–320, 2013.



Xiang Yin (S'14–M'17) was born in Anhui, China, in 1991. He received the B.Eng. degree from Zhejiang University, Hangzhou, China, in 2012, the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2013 and 2017, respectively, all in electrical engineering.

He is currently with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. His current research interests include supervisory control of discrete-event systems, and model-based fault diagnosis, formal methods, security and their applications to cyber and cyber-physical systems.

Dr. Yin was a recipient of the Outstanding Reviewer Award from *Automatica* in 2016, the Outstanding Reviewer Award from the IEEE TRANSACTIONS ON AUTOMATIC CONTROL in 2017 and the IEEE Conference on Decision and Control Best Student Paper Award Finalist in 2016. He is the Co-Chair of the IEEE CSS Technical Committee on Discrete Event Systems.



Zhaojian Li (S'15–M'16) received the B.S. degree in civil aviation from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2010, and the M.S. and Ph.D. degrees from the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA, in 2014 and 2016, respectively.

From 2010 to 2012, he was an Air Traffic Controller with the Shanghai Area Control Center, Shanghai, China. From 2014 and 2015, he was an Intern with Ford Motor Company, Dearborn, MI, USA. Since 2013, he has been a Graduate Research Assistant with the Department of Aerospace Engineering, University of Michigan. His current research interests include optimal control, system modeling, estimation, and intelligent transportation systems.

Dr. Li was a recipient of the National Scholarship from China.